ALGORITMIA Y COMPLEJIDAD

LABORATORIO - TEMA 4

EJERCICIO 1

Se dispone de una matriz M de tamaño $F \times C$ (F es la cantidad de filas y C la cantidad de columnas), cuyas celdas tienen un valor numérico entero positivo. Por ejemplo, la siguiente matriz con 4 filas y 5 columnas:

3	2	3	2	5
1	2	4	5	6
4	3	6	2	4
2	5	3	4	3

Un movimiento entre las casillas M_{ij} y M_{pq} es válido solamente si $\{p = i, q = j + 1\}$ o bien si $\{p = i + 1, q = j\}$. Se denomina camino de la matriz como la sucesión de movimientos que llevan de la casilla M_{11} a la casilla M_{FC} y el coste de un camino es igual a la suma de los valores de las casillas que recorre.

Diseñar un algoritmo de Programación Dinámica que obtenga el menor de los costes de todos los caminos de una matriz dada como parámetro.

EJERCICIO 2

Se quieren obtener los primeros N números primos (por ejemplo, si N = 5 habría que conseguir los primos 2, 3, 5, 7 y 11).

Diseñar un algoritmo de Programación Dinámica que resuelva el problema, encontrando una forma de reutilizar los resultados ya conseguidos para calcular los nuevos números primos.

EJERCICIO 3

Se tiene un sistema de billetes de distintos valores y ordenados de menor a mayor (por ejemplo 1, 2, 5, 10, 20, 50 y 100 euros), que se representan mediante los valores v_i, con $i \in \{1, \dots, N\}$ (en el caso anterior, $N = 7$) de manera que de cada billete se tiene una cantidad finita, mayor o igual a cero, que se guarda en c_i (siguiendo con el ejemplo, $c_3 = 6$ representaría que hay 6 billetes de 5 euros).

Se quiere pagar exactamente una cierta cantidad de dinero D, utilizando para ello la menor cantidad de billetes posible. Se sabe que $D \leq \sum_{i=1}^{n} c_i v_i$, pero puede que la cantidad exacta D no sea obtenible mediante los billetes disponibles.

Diseñar un algoritmo de Programación Dinámica que determine, teniendo como datos los valores c_i, v_i y D:

- Si la cantidad D puede devolverse exactamente o no.
- En caso afirmativo, cuantos billetes de cada tipo forman la descomposición óptima.

EJERCICIO 4

Alí Babá ha conseguido entrar en la cueva de los ciento un mil ladrones, y ha llevado consigo su camello junto con dos grandes alforjas; el problema es que se encuentra con tanto tesoro que no sabe ni qué llevarse. Los tesoros son joyas talladas, obras de arte, cerámica… es decir, son objetos únicos que no pueden partirse ya que entonces su valor se reduciría a cero.

Afortunadamente los ladrones tienen todo muy bien organizado y se encuentra con una lista de todos los tesoros que hay en la cueva, donde se refleja el peso de cada pieza y su valor en el mercado de Damasco. Por su parte, Alí sabe la capacidad de peso que tiene cada una de las alforjas.

Diseñar un algoritmo de Programación Dinámica que, teniendo como datos los pesos y valor de las piezas, y la capacidad de las dos alforjas, permita obtener el máximo beneficio que podrá sacar Alí Babá de la cueva de las maravillas.

EJERCICIO 5

Se tiene un grafo dirigido $G = \langle N, A \rangle$, siendo $N = \{1, \dots, n\}$ el conjunto de nodos y $A \subseteq N \times N$ el conjunto de aristas. Sea M la matriz de adyacencia del grafo G, es decir, $M_{ij} = True$ si existe la arista $[i, j]$, y $M_{ij} = False$ en caso contrario.

Se está interesado en saber desde qué vértices se puede acceder a cualquier otro vértice (mediante un camino de cualquier longitud), utilizando el algoritmo de Warshall, por lo que:

- Hay que obtener una matriz de caminos C de manera que $C_{ij} = True$, si existe un camino (de cualquier longitud) entre los nodos i y j, y $C_{ij} = False$ si no existe forma de llegar de i a j.
- Para ello, hay que ir considerando con un enfoque Bottom-Up una cantidad mayor nodos. Se empieza intentando ir directamente de un nodo a otro, después se intenta encontrar los caminos que pueden utilizar el vértice 1, a continuación, los que pueden usar los vértices 1 y 2, después lo que se

consiguen con los vértices 1, 2 y 3, etc., hasta llegar a obtener los caminos que pueden utilizar todos los vértices de 1 hasta n.

Implementar un método de Programación Dinámica en los términos anteriores, que obtenga la matriz de existencia de caminos C de un grafo teniendo como datos la cantidad de nodos n y la matriz de adyacencia M.

EJERCICIO 6

¡Llega el torneo de EscobaBall, y más salvaje que nunca! Este año, en el Colegio de Magia y Hechicería han decidido que los cuatro equipos (Grifos, Serpientes, Cuervos y Tejones) jueguen en cada partido todos contra todos, y como siempre que ningún partido termine en empate. El torneo acabará cuando un mismo equipo haya ganado un total de N partidos (no necesariamente consecutivos).

El aprendiz de mago Javi Potter quiere apostar algo de dinero por su equipo, los Grifos, así que se dirige a la casa de apuestas de los gnomos para ver cuánto le darían si gana su equipo: sus ganancias serían iguales a la cantidad de dinero apostado dividido por la probabilidad de que el equipo gane el campeonato (por ejemplo, si los Grifos tuviesen un 50% de ganar el campeonato y Javi apuesta 10 monedas de oro, sus posibles ganancias serían $10 \ / \ 0.5 = 20$ monedas de oro; si tuviesen un 20% de ganar, el beneficio posible sería de $10 \ / \ 0.2 = 50$ monedas de oro (una mayor recompensa al ser más difícil de conseguir).

Para obtener esta probabilidad, la casa de apuestas tiene un Valor de Calidad asignado a cada equipo (que mide la habilidad de los jugadores, su motivación, etc., y que es un valor fijo para el equipo e independiente del partido que esté jugando) de manera que cuanto mayor es el VC de un equipo, más probabilidades tiene de ganar un partido. Por ejemplo, si los cuatro equipos tuviesen igual VC todos tendrían un 25% de ganar un partido. Si tres equipos tuviesen el mismo VC y el cuarto equipo tuviese el doble de esa cantidad, los primeros tendrían un 20% y el último un 40%. Como los partidos no pueden terminar en empate, la suma de las probabilidades siempre es el 100%.

Teniendo como datos los Valores de Calidad de los equipos, la cantidad de partidos N que debe ganar un equipo para conseguir ganar el torneo, y el dinero D apostado por Javi Potter, obtener cuáles serían las ganancias si ganasen los Grifos.

EJERCICIO 7

Se define una secuencia de bits A como una sucesión $A = \{a_1, a_2, \dots, a_n\}$ donde cada a_i puede tomar el valor 0 o 1. A partir de una secuencia se define una subsecuencia X de A como $X = \{x_1, x_2, \dots, x_k\}$, siendo $k \leq n$, de forma que X puede obtenerse eliminando algún elemento de A pero respetando el orden en que aparecen los bits; por ejemplo, si $A = \{1,0,1,1,0,0,1\}$ podríamos obtener como subsecuencias $\{1,1,1,0,1\}$, $\{1,0,1\}$ o $\{1,1,0,0\}$ entre otras, pero nunca se podría conseguir la subsecuencia $\{1,0,0,1,1\}$.

Dadas dos secuencias A y B, se denomina a X una subsecuencia común de A y B cuando X es subsecuencia de A y además es subsecuencia de B (aunque puede que se hayan obtenido quitando distintos elementos en A que B, e incluso distinta cantidad). Suponiendo las secuencias $A = \{0,1,1,0,1,0,1,0\}$ y $B = \{1,0,1,0,0,1,0,0,1\}$ una subsecuencia común sería $X = \{1,1,0,1\}$, pero no podría serlo $X = \{0,1,1,1,0\}$.

Se desea determinar la subsecuencia común de dos secuencias A y B que tenga la longitud máxima, para lo que se pide:

- Explicar con detalle la forma de resolver el problema.
- Hacer un algoritmo de Programación Dinámica que obtenga la longitud máxima posible y una secuencia común de dicha longitud.

Entregables: un ejercicio a elegir entre los problemas 3 y 4 y un ejercicio a elegir entre los problemas 6 y 7.